ARL-TR-7438 ● SEP 2015

US Army Research Laboratory

MUVES-S2 Adaptive Geometry User Guide

by Matthew C Rothwell and James Hunt

Approved fro public release; distribution is unlimited.

NOTICES

Disclaimers

ARL-TR-7438 ● SEP 2015

US Army Research Laboratory

MUVES-S2 Adaptive Geometry User Guide

by Matthew C Rothwell
Survivability/Lethality Analysis Directorate, ARL

James Hunt
Applied Research Associates, Inc., Aberdeen, MD

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2015	Final	1–30 June 2014

4. TITLE AND SUBTITLE
MUVES-S2 Adaptive Geometry User Guide

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Matthew C Rothwell and James Hunt

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-SLB-G
Aberdeen Proving Ground, MD 21005-5068

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-7438

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The ability to conduct analyses using adaptive geometry was funded as part of the 2014 Tactical Tools Techniques and Methodology (TTTM) program. The development of this capability was spurred by the US Army Research Laboratory's (ARL's) Weapons and Materials Research Directorate focus on developing adaptive protection systems. To enable ARL's Survivability/Lethality Analysis Directorate (SLAD) to conduct analyses of these newly developed technologies, SLAD funded the adaptive geometry TTTM proposal. The new adaptive geometry methodology provides for the analysis of variable target configurations and scene-based analyses without requiring manual target geometry modification.

15. SUBJECT TERMS
MUVES-S2, adaptive geometry, scene, vulnerability analysis, TTTM

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
					Matthew C Rothwell
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER (Include area code)
Unclassified	Unclassified	Unclassified	UU	26	410-278-5476

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

INTENTIONALLY LEFT BLANK.

1. Introduction

The ability to conduct analyses using adaptive geometry was funded as part of the 2014 Tactical Tools Techniques and Methodology (TTTM) program. The development of this capability was spurred by the US Army Research Laboratory's (ARL's) Weapons and Materials Research Directorate focus on developing adaptive protection systems. The new adaptive geometry methodology provides for the analysis of variable target configurations and scene-based analyses without requiring manual target geometry modification.

The Survivability/Lethality Analyisis Directorate's Software Development and the Ground Mobile Branches worked to develop the adaptive geometry capability that was released in MUVES-S2 version 2.39 for use by analysts. This document will review the requirements needed for BRL-CAD and MUVES-S2 users to conduct an analysis using the adaptive geometry capability. Additional information on the use of the adaptive geometry can be found in the MUVES-S2 Change Design Document for System Change Request No.1845.[1]

2. Current Analysis Approach

The current approach to analyzing vehicles that have multiple geometric configurations is to create a unique geometric model for each configuration. The modeling effort required is minimal to create a few configurations, for example, the launch and travel models of a missile launcher. However, it is not feasible to create individual target descriptions when analyzing adaptive armor configurations, which may have hundreds of possible configurations. The adaptive geometry TTTM creates a solution within the MUVES-S2 analysis software, while also enhancing MUVES-S2 capabilities by providing the ability to conduct analysis of multitarget scenes.

3. BRL-CAD Requirements

When creating a BRL-CAD target for use in adaptive geometry the following 2 key requirements must be followed:

- Each object to be translated or rotated requires a top-level group (e.g., hull, doors, hatches, turret, and armor panel). Each top-level group must be independent of each other (i.e., the top-level hull group should not include the doors, hatches, turret, or armor panels).

- The origin of the components to be translated or rotated should be identified for use by the MUVES-S2 analyst.

For example, the high-mobility multipurpose wheeled vehicle (HMMWV) with gunner protection kit (GPK) and 4 doors would have 6 top-level groups:

- HMMWV (0 mm, 0 mm, 0 mm): front bumper centerline on ground plane
- GPK (–1,400 mm, 0 mm, 0 mm): rotate about Z axis center on turret ring
- Front_left_door (–1,200 mm, 1,142.7 mm, 0 mm): rotate about Z axis on door hinges
- Front_right_door (–1,200 mm, –1,142.7 mm, 0 mm): rotate about Z axis on door hinges
- Rear_left_door (–1,500 mm, 1,142.7 mm, 0 mm): rotate about Z axis on door hinges
- Rear_right_door (–1,500 mm, 1,142.7 mm, 0 mm): rotate about Z axis on door hinges

Currently there are no constraints limiting the range of motion of components. This could result in multiple overlapping components when creating a scene. When translating/rotating components during an analysis the MUVES-S2 analyst must be cognizant to not allowing components to intersect or overlap.

4. Adaptive Geometry Approach

When defining a set of target input files, the user must be able to specify multiple "geometry" specifications in the target directory. This will allow for reusing target geometry and associated inputs for different top-level objects in the geometric model.

Examples:

- An existing tank could be broken into hull and turret targets allowing the turret to be placed in the scene relative to the hull but at different turret azimuth angles.
- Personnel could be separated from the top-level object and placed into the scene at different vehicle locations or dismounted.
- Armor could be separated from the top-level object and placed at different locations and/or angles to analyze the effect on vehicle vulnerability.

Traditionally, the MUVES-S2 session file was used to define the input files needed to conduct an analysis. To conduct analyses using the adaptive geometry methodology, the addition of a scene file was required. The session and scene files will now be used to completely define all required files. The scene file will consist of multiple target groups, each describing the portions of the vehicle and if any manipulation is to occur. The session file will define which scenes are to be analyzed and if any line drawings are to be created to confirm that the intent of the analysis is met.

Scene File Syntax

Like most other MUVES-S2 input files, the new scene file can contain comments on any line starting with the # character.

Target Geometries

Each target is specified with the "target" keyword, followed by a unique target name and open curly brace ({) character. The target name may not contain space or tab characters but may use any other printable characters, for example:

> target t62 {

The scene file will define the location of the following files:

- **target**: Specifies the name of the target directory containing the geometric model database and optional "geometry" file. This keyword is required.
- **states**: Specifies the name of the target's state vectors file. If the target has no critical components or systems, this keyword is not needed.
- **sysdef**: Specifies the name of the target's system definition file. If the target has no critical systems, this keyword is not needed.
- **category**: Specifies the name of the component category file. This keyword is required.
- **property**: Specifies the name of the component properties file. This keyword is not required, but most targets cannot be analyzed without it.
- **matprop**: Specifies the name of the material properties file. This keyword is not required if the default material properties will be used.
- **eval**: Specifies the name of the damage evaluation selection file. If the target has no critical systems, this keyword is not needed.

- **evaldata**: Specifies the name of the evaluation data (i.e., “ecurve”) file. If the target has no critical components that need evaluation tables, this keyword is not needed.
- **interdata**: Specifies the name of the interaction data (i.e., “icurve”) file. If the target has no components that need interaction tables, this keyword is not needed.
- **bcurve**: Specifies the name of the blast curve data file. If the target has no blast tables, this keyword is not needed.
- **params**: Specifies the name of the target’s parameters file. If the target has no components that need additional parameters, this keyword is not needed.
- **packages**: Specifies the name of the target’s armor package definitions file. If the target has no armor packs, then this file is not needed.
- **geometry**: Specifies the start of a block of input containing keywords normally found in the “geometry” file in the target directory. If this block is defined, the “geometry” file in the target directory is ignored. If this block is not defined, a “geometry” file is required and used in the target directory. An open curly brace ({) character is required after the geometry keyword with whitespace (space or tab character) in between. A close curly brace (}) character on a line by itself is required to complete the geometry block.

A close curly brace (}) character on a line by itself is required to complete a target definition block.

Geometry Instances

After the input files are defined, all instances are specified using the “instance” keyword, followed by a unique instance name, reference to a previously defined target name, and open curly brace ({) character. The instance name may not contain space or tab characters or period (.) character but may use any other printable characters. The period (.) character is used to specify and output components and systems at the scene level (e.g., “HMMWV1.gunner”). The following is an example of the start of an instance definition block:

```
instance t62_number1 t62 {
```

The manipulation of multiple geometry regions/instances will be controlled by translation or rotation about an origin set for each top-level group.

- **translation**: Specifies the coordinate translation of the target from the model's local coordinate system to the scene’s world coordinate system. The units are specified in millimeters, and the keyword is not required if no translation is needed. For example, to position a vehicle 20 m to the left of its model origin:

translation 0 20000 0

- **origin**: Specifies the rotation origin of the target instance in the target model's local coordinate system. The units are specified in millimeters, and the keyword is not required if no rotation is needed. This keyword is also not needed if the target is to be rotated about the local coordinate origin (0, 0, 0). For example, to rotate the door about the hinge pin, provided in the door top level group (–1,200 mm, 1,142.7 mm, 0 mm) rather than the local coordinate origin:

  origin –1200 1142.7 0

Rotation is controlled with one of the following 7 rotation angles keywords:

- **azel**: Specifies the azimuth and elevation rotation angles of rotation in degrees. Positively applied right-hand rule rotations are performed with azimuth first (about the Z axis) followed by elevation (about the rotated Y axis).
- **rpy**: Specifies the roll, pitch, and yaw rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of roll (about X axis), pitch (about rotated Y axis), and then yaw (about rotated Z axis).
- **pyr**: Specifies the pitch, yaw, and roll rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of pitch (about Y axis), yaw (about rotated Z axis), and then roll (about rotated X axis).
- **ypr**: Specifies the yaw, pitch, and roll rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of yaw (about Z axis), pitch (about rotated Y axis), and then roll (about rotated X axis).
- **ryp**: Specifies the roll, yaw, and pitch rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of roll (about X axis), yaw (about rotated Z axis), and then pitch (about rotated Y axis).
- **yrp**: Specifies the yaw, roll, and pitch rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of yaw (about Z axis), roll (about rotated X axis), and then pitch (about rotated Y axis).
- **pry**: Specifies the pitch, roll, and yaw rotation angles in degrees. Positively applied right-hand rule rotations are applied in the order of pitch (about Y axis), roll (about rotated X axis), and then yaw (about rotated Z axis).

Rotations about the locally specified origin are performed before translations when going from local to world coordinates. A close curly brace (}) character on a line by itself is required to complete an instance definition block. For example:

```
instance t62.number1 t62 {
        translation 0 20000 0
        azel –30 20
}
```

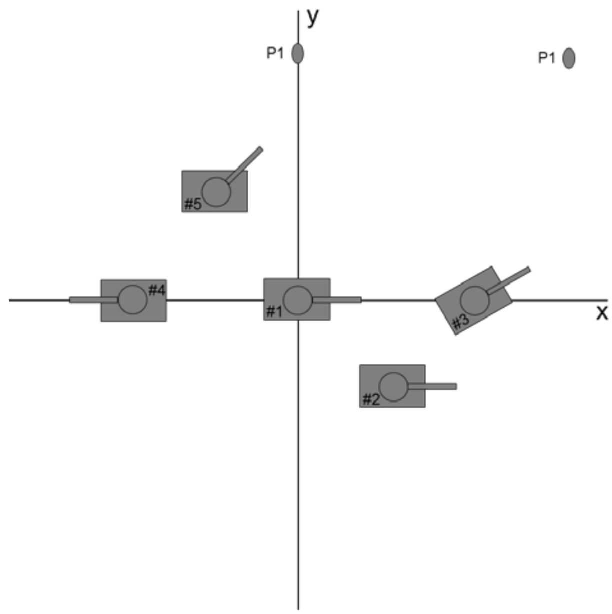
The translation and rotation parameters can be controlled using basic math operators surrounded by parentheses which allows for multiple values to be entered as a part of the session file.

Example Scene File: 5 vehicles and 2 dismounted personnel

```
target t62 {
        category inputs/ccmap/t62.s2
        property inputs/prop/t62.s2
        eval inputs/des/t62.s2
        evaldata inputs/ecurve/t62.s2
        matprop inputs/matprop/t62.s2
        params inputs/params/t62.s2
        states inputs/states/t62.s2
        system inputs/sysdef/t62.s2
        target inputs/target/t62.s2
}
# geometry file located in target folder is used
```

or

```
target t62.hull {
        category inputs/ccmap/t62.s2
        property inputs/prop/t62.s2
        eval inputs/des/t62.s2
        evaldata inputs/ecurve/t62.s2
        matprop inputs/matprop/t62.s2
        params inputs/params/t62.s2
        states inputs/states/t62.s2
        system inputs/sysdef/t62.s2
        target inputs/target/t62.s2
        geometry {
                method          mged
                mged_database   T62.g
                root_object     hull
                region_map      regions.hull
                }
```




```
}
target t62.turret {
        category inputs/ccmap/t62.s2
        property inputs/prop/t62.s2
        eval inputs/des/t62.s2
        evaldata inputs/ecurve/t62.s2
        matprop inputs/matprop/t62.s2
        params inputs/params/t62.s2
        states inputs/states/t62.s2
        system inputs/sysdef/t62.s2
        target inputs/target/t62.s2
        geometry {
                method                  mged
                mged_database           T62.g
                root_object     turret
                region_map      regions.turret
                }
}

target orcaman {
        category
inputs/ccmap/ccmap.orcaman
        property inputs/prop/prop.orcaman
        eval inputs/des/des.orcaman
        matprop inputs/matprop/matprop.orcaman
        params inputs/params/params.orcaman
        states inputs/states/states.orcaman
        system inputs/sysdef/sysdef.orcaman
        target inputs/target/StandingORCAman
   }
# geometry file located in target folder is used


instance tank1 t62 {
}
instance tank2 t62 {
        translation 5000 -5000 0
}
instance tank3 t62 {
        translation 11000 0 0
        azel -30 0
}
instance tank4 t62 {
        translation -11000 0 0
        rpy 0 0 180
}
```

```
instance tank5_hull t62_hull {
        translation -5000 5000 0
}
instance tank5_turret t62_turret {
        translation -5000 5000 0
azel -45 0
}
instance troop1 orcaman {
        translation 0 5000 0
        rpy ($2) ($3) ($4)
        # "$" denotes a scene paramater variable defined in the session file, this is
defined in further detail in the session file section below and in appendix A.
}
instance troop2 orcaman {
        translation ($1 * 1000) 5000 0
        ryp ($2) ($4) ($3)
        # "$" denotes a scene paramater variable defined in the session file, this is
defined in further detail in the session file section below and in appendix A.
        file
}
```

Environment Variable

For each view/scene combination, a line drawing of the scene can be produced by specifying the MUVES_DRAW_SCENE_SCALE environment variable. For example:

```
env MUVES_DRAW_SCENE_SCALE 0.5
```

The final results file name will be used as the base name for the image file with the scene parameters, azimuth, and elevation, and ".bw" appended as a suffix. For example, a scene with 3 parameters (2,000, 7.5, and –14) and 35, 25 azimuth, elevation shot specification would get an image file with a name "results/1.0.fr.2000,7.5,-14.35,25.bw". If a file by that name already exists, it will not be overwritten and line drawing will be skipped. Therefore, the files should manually be removed if the scale, geometry models, or scene files are modified.

Session file

One or more scenes can be specified in a session file run using one or more scene files. Each scene will be processed in the order specified and will create one or more view results for each combination of view and threat specification. On each scene specification line in the session file, additional scene parameters may be specified that can be used with any keyword in instance definition blocks. Any number of scene parameters can be specified in the session file. Each parameter is indexed and can be used within the instance using a dollar sign (“$”) character followed by the scene parameter index. For example, if the session file contains the scenes:

```
scene inputs/scene/scene.t62 1000 20 15
scene inputs/scene/scene.t62 3000 10 30
```

and the scene file contains an instance rotation specification:

```
rpy ($3) 0 0
```

then the same scene will be analyzed twice with the instance rotated 15° the first time and 30° the second time. Scene parameters must be surrounded by parentheses and may be combined with the basic arithmetic operators of addition (+), subtraction (–), multiplication (*), and division (/). For example, if the previous instance rotation was

```
rpy (2 * $3) 0 0
```

the instance would be rotation 30° the first time and 60° the second time.

Sequences of scene parameters can be specified on the scene line in the session file using start, stop, and step values separated by 2 colon characters (:) and no spaces or tabs. For example, to specify 5 scenes using a scene parameter with the values 20, 25, 30, 35, and 40:

```
scene inputs/scene/scene.hmmwv_rotations 20:40:5
```

If multiple sequences are specified for multiple scene parameters, each combination will be analyzed. For example, the line

```
scene inputs/scene/scene.hmmwv_locations 0:4:1 -2:2:2
```

would produce 15 scenes using combinations of 0, 1, 2, 3, and 4 for the first scene parameter and –2, 0, 2 for the second scene parameter.

Example Session File: Modifications needed for adaptive geometry are highlighted in gray.

```
title sample_scene
classification inputs/classif/IsExampleUnclassified.xml
approx s2
env MUVES_DRAW_SCENE_SCALE 0.5
modkey ke_pen KE_LOS
modkey orca_penetration ON
threat inputs/threat/sample_ke range 100
scene inputs/scene/scene.t62s 1000 0 10 17
scene inputs/scene/scene.t62s 1000 0.0035 0 -1.8
```

```
scene inputs/scene/scene.t62s 3:5:2 0 0 -90:90:45
vector troop0.ORCA_man_injury
vector troop1.ORCA_man_injury
vector tank1.DAL_kills
vector tank2.DAL_kills
vector tank3.DAL_kills
vector tank4.DAL_kills
viewfile inputs/view/scene.view
analyze results/1.0.fr sessions/1
```

5. Summary and Conclusion

The adaptive geometry TTTM provides the capability to conduct analyses that are based on easily manipulated geometry, allowing for the analysis of complex scenes. However, when using this capability, the analyst must be cautious to not induce errors by creating overlapping components. At the release of the adaptive geometry capability, there are no constraints limiting the range of motion of components. This could result in multiple overlapping components within a target or scene. For example, a target could be analyzed with a door rotated more than 180° about its pivot axis without causing the MUVES-S2 code to terminate. When translating/rotating components during an analysis, the MUVES-S2 analyst must be cognizant to not allow components to intersect or overlap. Future recommendations are to create equations of constraint within the CAD geometry that will remove the potential for component overlap.

A PowerPoint presentation and sample analysis files have been created to explain the analysis changes needed to use the adaptive geometry capability. The presentation slides can be found in the Appendix, and sample analyses can be found on the following unclassified and classified networks:

Unclassified: \\hera\muves\2.41\data\analysis\samples

Session 39

Classified: \\royal\j\vsblast\muves\analysis\M1151_Adaptive_TTTM
Sessions 99:102, 500, 600

6. Reference

1. Hunt J. MUVES–S2 Change design document for System Change Request (SCR) No. 1845; ability to rotate/translate portions of the target description; Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2014 Aug 26.

INTENTIONALLY LEFT BLANK.

Appendix. Adaptive Geometry PowerPoint Presentation

This appendix appears in its original form, without editorial change.

UNCLASSIFIED

Adaptive Geometry

Matthew Rothwell
Matthew.C.Rothwell.civ@mail.mil
410-278-5476

Jim Hunt
James.E.Hunt7.ctr@mail.mil
410-278-7959

Overview

Steps to conduct an adaptive geometry analysis:

- Target description with multiple top level groups.
- Scene file defining the targets to be analyzed and configuration.
- Environment variable scaling for line drawings.
- Session file defining manipulations which are to occur.

Results:

- Analysis results for manipulated geometry
- Analysis results for multi-target scenes
- Line drawings for each view/scene combination

Target Description

When creating CAD for use in adaptive geometry two key requirements must be followed:

1. For each object to be translated or rotated a top level group must exist (example: doors, hatches, turret, armor panel)
2. The origin of the object that is to be translated/rotated should be identified when the CAD is delivered. (example: hinge axis, turret ring, main gun)

What is needed

- .g file with multiple top level groups for each object to be manipulated
- Readme file with origin and rotation axis of objects

For example to conduct an analysis which included opening and closing all doors and slewing the GPK the readme file would read as such:

- HMMWV (0mm, 0mm, 0mm): front bumper centerline on ground plane
- GPK (-1400mm, 0mm, 0mm): Z rotation axis centered on turret ring
- Front_left_door (-1200mm, 1142.7mm, 0mm): Z rotation axis on door hinges
- Front_right_door (-1200mm, -1142.7mm, 0mm): Z rotation axis on door hinges
- Rear_left_door (-1500mm, 1142.7mm, 0mm): Z rotation axis on door hinges
- Rear_right_door (-1500mm, -1142.7mm, 0mm): Z rotation axis on door hinges

Additional objects to manipulate would require additional top level groups.

Note: At this time there are no constraints limiting the range of motion of components. This could result in multiple overlapping components within a scene. When translating/rotating components during an analysis the MUVES-S2 analyst must be cognizant to not allow components to intersect or overlap.

Scene File: Targets

The addition of a scene file allows for the manipulation of geometry and the analysis of multiple targets configured within a scene. First, all targets/objects must be identified.

The scene file in combination with the session file define required input files for the analysis

What is needed:

- location of the following files for each target in your analysis:
  - ccmap
  - prop
  - des
  - ecurve
  - matprop
  - params
  - states
  - system
  - target
  - geometry

Example inputs for a two target scene:

```
target t62 {
        category inputs/ccmap/t62.s2
        property inputs/prop/t62.s2
        eval inputs/des/t62.s2
        evaldata inputs/ecurve/t62.s2
        matprop inputs/matprop/t62.s2
        params inputs/params/t62.s2
        states inputs/states/t62.s2
        system inputs/sysdef/t62.s2
        target inputs/target/t62.s2
        geometry {
                method          mged
                mged_database   T62.g
                root_object     hull
                region_map      regions.hull
                }
        }
target orcaman {
        category inputs/ccmap/ccmap.orcaman
        property inputs/prop/prop.orcaman
        eval inputs/des/des.orcaman
        matprop inputs/matprop/matprop.orcaman
        params inputs/params/params.orcaman
        states inputs/states/states.orcaman
        system inputs/sysdef/sysdef.orcaman
        target inputs/target/StandingORCAman
        }
```


Scene File: Translation/Rotation

The second portion of the scene file defines what translations/rotations are to occur. This is where multiple objects can be added and manipulated to create a target with articulated regions or a scene with multiple targets.

What is needed:

- Information on the analysis scene:
  - Sytargets tems to be analyzed (instances)
    Note: These could be entire vehicles or individual parts such as hull, turret, and hatches.

    instance unique_name group_from_.g {

  - **Translation** -specifies the coordinate translation of the target

    translation X Y Z

  - **Origin** -specifies the rotation point of the target instance defined in the CAD readme file. Is not needed if the target is to be rotated about the local coordinate origin (0, 0, 0)

    origin X Y Z

- Rotations are specified using one of the following key words for azimuth and elevation or roll, pitch and yaw in degrees.
  - **azel**
  - **rpy**
  - **pyr**
  - **ypr**
  - **ryp**
  - **yrp**
  - **pry**

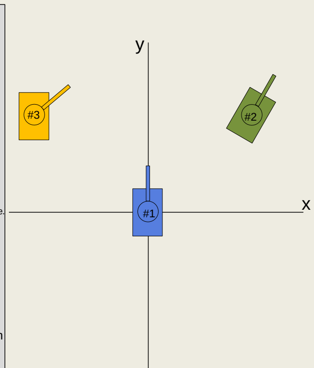
azel Az El
or
rpy R P Y

Example Scene

Example scene which can be analyzed as a direct or indirect fire scenario.

```
#Tank #1 with no translation or rotation:
instance tank1 t62 {
        }

#Tank #2 with a translation and rotation:
instance tank2 t62 {
        translation 11000 100 0
        azel 30 0
        }

#Tank #3 with a translation and rotated turret about a defined origin:
instance tank3_hull t62.hull {
        translation -11000 ($1) 0     # uses 1st field of the scene line in the session file.
        }

instance tank3_turret t62.turret {
        translation -11000 ($1) 0
        origin 10 20 0      # from target geometry read me file or user defined
        ypr ($2) 0 0        # uses 2nd field of the scene line in the session file.
        }
```

Note: Each translation/rotation parameter is indexed and can be used in the instance definition using a dollar sign ('$') character which will be defined in the session file. This is useful when conducting analyses on multiple configurations (example: turret rotated at 5° increments from 0° to 180°).



UNCLASSIFIED

Environment Variable:

Using an environment variable allows for the creation of line drawings to ease in the visualization of the scene. For each view/scene combination a line drawing will be produced by using the MUVES_DRAW_SCENE_SCALE environment variable. This image file will have a .bw extension and be located in the results directory.

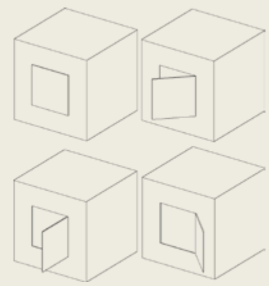
What is needed:

- User defined image scale factor

env MUVES_DRAW_SCENE_SCALE scale_factor

Example:

env MUVES_DRAW_SCENE_SCALE 0.5



UNCLASSIFIED The Nation's Premier Laboratory for Land Forces

UNCLASSIFIED

Session File:

The session file now has the ability to conduct analyses of multiple scenes within one analysis. The dollar sign ('$') character which was used in the scene file is defined here.

What is needed:

- Environment variable
- Scene file(s)
  - Multiple scene file can be analyzed in one analysis
  - The analysis will run through all scene files in order and if multiple parameters are specified all combinations will be conducted.

Example Session file:

Note: Modifications needed for adaptive geometry are highlighted in green

```
title sample_scene
classification inputs/classif/IsExampleUnclassified.xml
approx s2
env MUVES_DRAW_SCENE_SCALE 0.5
modkey ke_pen KE_LOS
modkey orca_penetration ON
threat inputs/threat/sample_ke range 100
scene inputs/scene/scene.t62s
scene inputs/scene/scene.t62s 100 50
scene inputs/scene/scene.t62s 100 0:45:5
vector troop0.ORCA_man_injury
vector troop1.ORCA_man_injury
vector tank1.DAL_kills
vector tank2.DAL_kills
vector tank3.DAL_kills
vector tank4.DAL_kills
viewfile inputs/view/scene.view
analyze results/1.0.fr sessions/1
```

UNCLASSIFIED The Nation's Premier Laboratory for Land Forces

UNCLASSIFIED

Example: #1

Scene:

```
target BOX{
        category inputs/ccmap/box_ccmap
        property inputs/prop/box_prop
        eval inputs/des/box_des
        evaldata inputs/ecurve/box_ecurve
        matprop inputs/matprop/box_matprop
        params inputs/params/box_params
        states inputs/states/box_states
        system inputs/sysdef/box_sysdef
        target inputs/target/box_target
        geometry {
                method          mged
                mged_database   boxscene.g
                root_object     box
                region_map      regions
                }
        }
target DOOR{
        category inputs/ccmap/box_ccmap
        property inputs/prop/box_prop
        eval inputs/des/box_des
        evaldata inputs/ecurve/box_ecurve
        matprop inputs/matprop/box_matprop
        params inputs/params/box_params
        states inputs/states/box_states
        system inputs/sysdef/box_sysdef
        target inputs/target/box_target
        geometry {
                method          mged
                mged_database   boxscene.g
                root_object     door
                region_map      regions
                }
        }

#box with a translation:
instance boxinstance BOX {
        translation 0 -100 0
        }
#door with a translation and rotation about origin:
instance doorinstance DOOR{
        translation 0 -100 0
        origin 0 -50 50
        ypr ($1) 0 0
        }
```

Readme file:

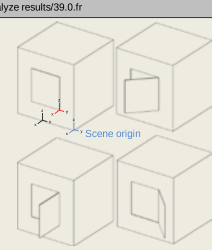
Box (0mm, 0mm, 0mm): center below door on ground plane

Door (0mm, 50mm, 50mm): Z rotation axis centered on hinge

Session:

```
title sample scene test case for SCR 1845
classification inputs/classif/IsExampleUnclassified.xml
approx s2
env MUVES_DRAW_SCENE_SCALE 0.5
scene inputs/scene/boxscene 0:135:45
viewfile inputs/view/boxscene.view
vector BoxInstance.Residual
modkey ke_pen KE_LOS
threat inputs/threat/sample_ke range 100
#intermediate results/39.0.ir
analyze results/39.0.fr
```

$1 field



UNCLASSIFIED The Nation's Premier Laboratory for Land Forces

Example: #2

Readme file:

Hull (0mm, 0mm, 0mm): centerline of vehicles front bumper on ground plane
Turret (-2000mm, 0mm, 1500mm): Z rotation axis centered on turret ring

Scene:

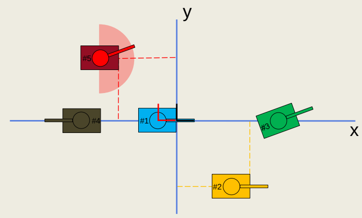
```
#Scene origin (0, 0, 0)
target t62 {
        category inputs/ccmap/t62.s2
                .....}
}

instance tank1 t62 {
}
instance tank2 t62 {
                translation ($1) (-$1) 0
# "$" denotes a scene parameter variable defined in the session file
}
instance tank3 t62 {
                translation 10000 0 0
                azel -20 0
}
instance tank4 t62 {
                translation -10000 0 0
                rpy 0 0 180
}
instance tank5_hull t62_hull {
                translation (-$1) ($1) 0
}
instance tank5_turret t62_turret {
                translation (-$1) ($1) 0
                origin -2000 0 1500
                azel ($2) 0
}
```

Session:

```
title sample_scene
classification inputs/classif/IsExampleUnclassified.xml
approx s2
env MUVES_DRAW_SCENE_SCALE 0.5
modkey ke_pen KE_LOS
modkey orca_penetration ON
threat inputs/threat/sample_ke range 100
scene inputs/scene/scene.t62s 5000 -20
scene inputs/scene/scene.t62s 5000 -90:90:45
vector tank1.DAL_kills
        .
        .
        .
analyze results/1.0.fr sessions/1
```

$1 field

$2 field



Sample Analyses

Sample analyses can be found on:

Unclassified: \\hera\muves\2.41\data\analysis\samples
Session 39

Classified: \\royal\j\vsblast\muves\analysis\M1151_Adaptive_TTTM
Sessions 99:102, 500, 600

INTENTIONALLY LEFT BLANK.

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL IMAL HRA MAIL & RECORDS MGMT
1 (HC)	DIR US ARMY EVALUATION CTR HQ TEAE SV P A THOMPSON 2202 ABERDEEN BLVD 2ND FL APG MD 21005-5001
6 (1 HC, 5 PDF)	DIR USARL RDRL SL P BAKER (1 PDF) G KUCINSKI (1 PDF) RDRL SLB G P MERGLER (1 PDF) M PERRY (1 PDF) M ROTHWELL (1 HC, 1 PDF)

INTENTIONALLY LEFT BLANK.